Left

Practical Animation of Liquids

Nick Foster
PDI-DreamWorks

Ronald Fedkiw
Stanford University

Goal

Build a 3D liquid simulator suitable for direct use as an animation tool

- Smooth believable surface that obeys the physics of liquids
- Interaction with animated objects and environments
- Efficient relative to other CG components

Liquid Behavior

- Surface motion highly coupled to events within the whole volume
- Incompressibility gives distinctive “sloppy” appearance (also numerical stiffness)
- Pressure waves move faster than velocity waves and cannot be ignored
- Surface can separate and reform while always remaining smooth and continuous

Related Work

Volumetric Techniques

- Miller and Pearce (1989)
- Foster and Metaxas (1996)
- Desbrun and Cani-Gascuel (1998)
- Stam (1999)

Computational Fluid Dynamics

- Fedkiw, *et al* (1999)

Incompressible Navier-Stokes

Conservation of momentum

$$\frac{\partial \mathbf{u}}{\partial t} = \nu \nabla \cdot (\nabla \mathbf{u}) - (\mathbf{u} \cdot \nabla)\mathbf{u} - \frac{1}{\rho}\nabla p + \mathbf{g}$$

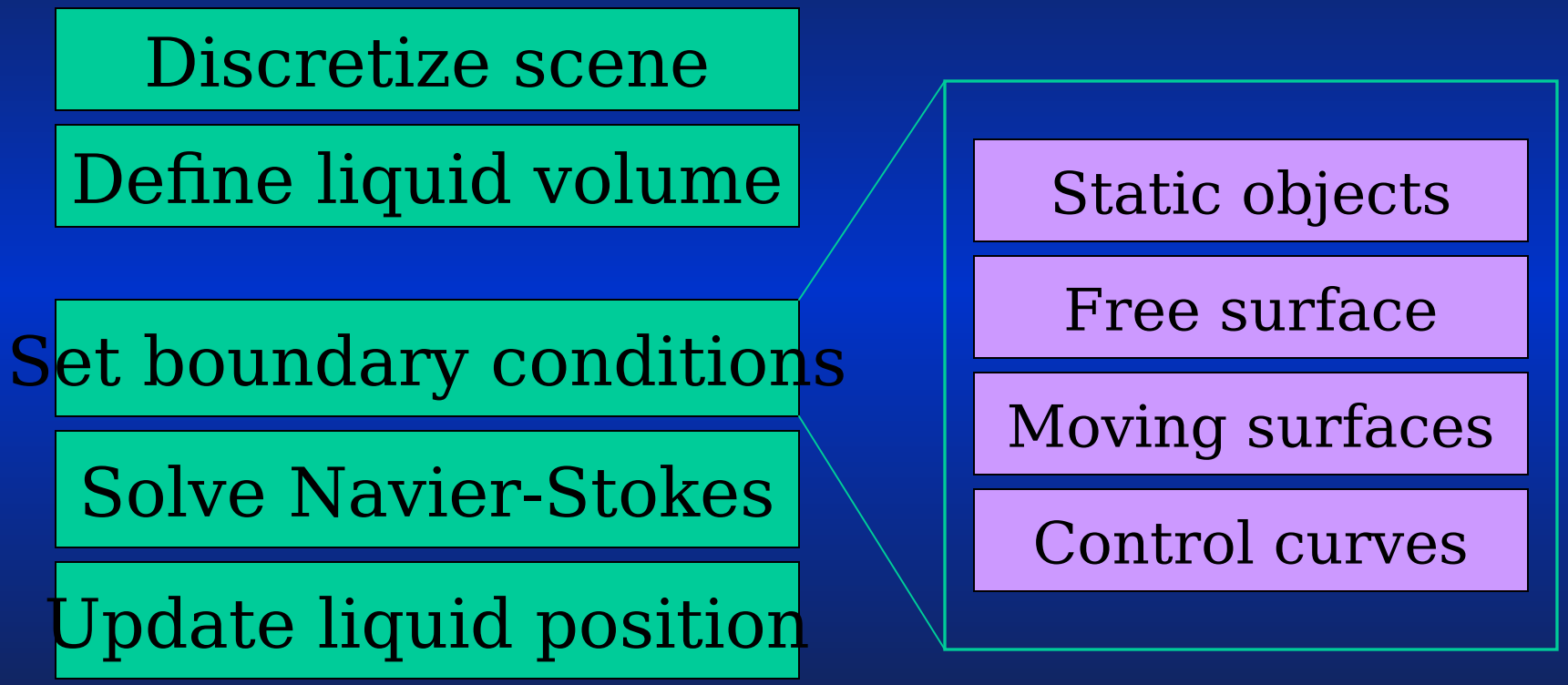
viscosity convection pressure

Conservation of mass

$$\nabla \cdot \mathbf{u} = 0$$

$$\nabla = \left\{ \partial / \partial x, \partial / \partial y, \partial / \partial z \right\}$$

Animation Pipeline



Integrating the Equations

First order semi-Lagrangian method on troublesome convective component

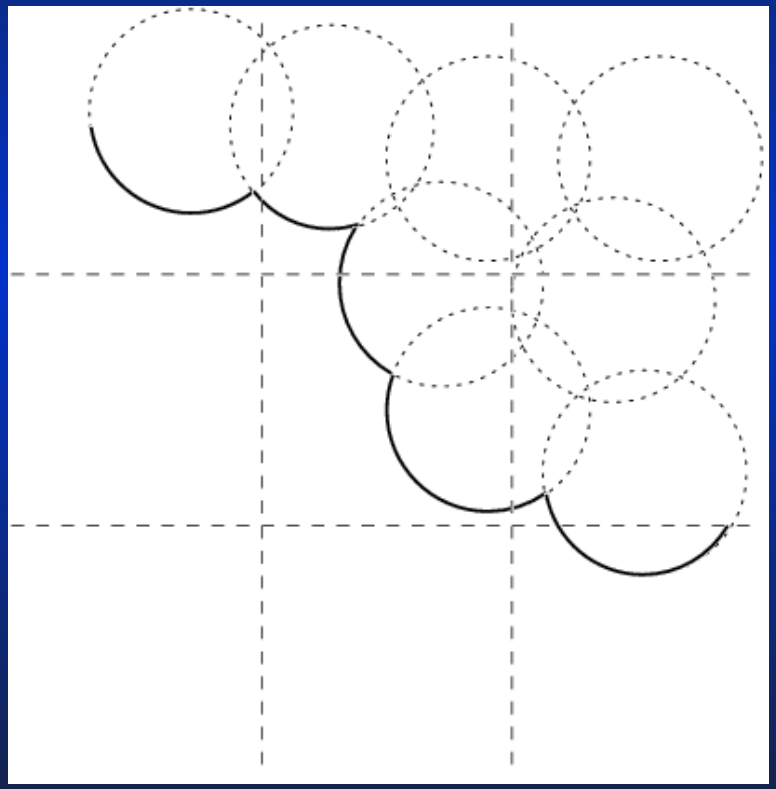
$$(\mathbf{u}\cdot\nabla)\mathbf{u}$$

Standard central differences on remaining terms (Foster and Metaxas 1997)

Representing the Surface

Particles

- Lagrangian approach
- Sub-grid resolution
- Retains detail
- Surface artifacts need smoothing
- Requires high density of particles



Moving the Surface

We need to update the potential field so that it obeys the physics of fluids

$$\frac{d\varphi}{dt} + \mathbf{u} \cdot \nabla\varphi = 0$$

Final field modified by particle position (using surface curvature metric) and locally

smoothed

Surface Rendering

- Fast Marching Method (Sethian 1999) used to build signed distance function
- Render surface directly
- Surface normal given by

$$\mathbf{n} = \frac{\nabla \varphi}{|\nabla \varphi|}$$

Integrating Moving Objects

1. Set intersecting cells to have object velocity
2. Apply Navier-Stokes as if object is liquid
3. Explicitly set velocity normal to object
4. Set all internal cells back to object velocity
5. Run mass conservation step as usual

Abusing the Velocity Field

Velocities within the computational grid can be set directly for whatever twisted purpose you like, provided that

- They satisfy the CFL condition
- You apply them using the moving object mechanism

Limited and imperfect control

Summary

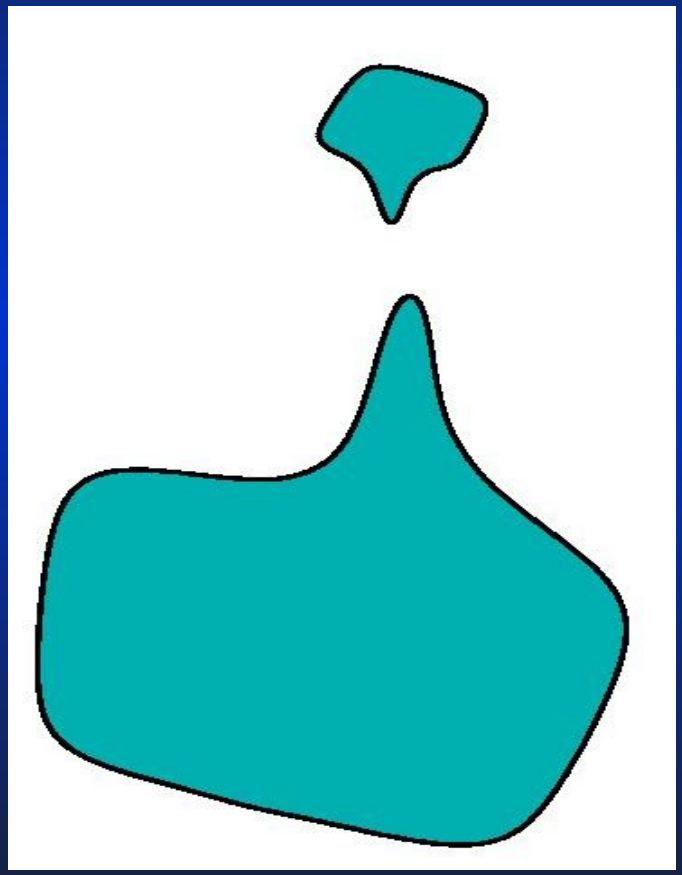
- Full 3D liquid motion, with volume preserving splashing
- Temporally and spatially smooth surface
- Interaction (one-way) with animated objects
- General low-level control mechanism (limited effects)
- Good enough for an integrated animation package

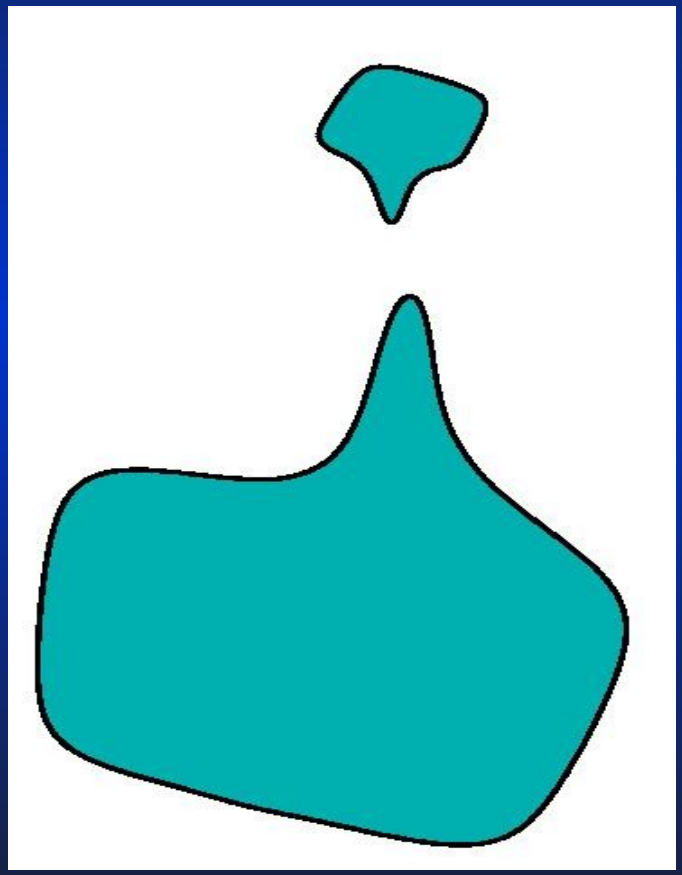
Right

Contents

Concentrating on the novel aspects of our approach

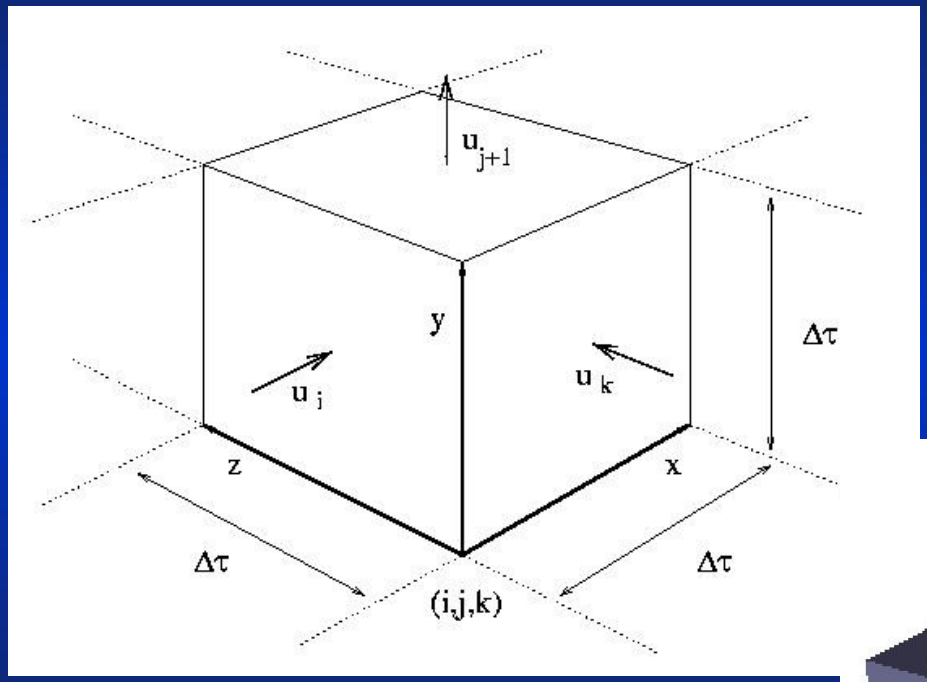
- Basic animation engine
- Tracking and rendering a smooth surface
- Interaction with polygonal objects

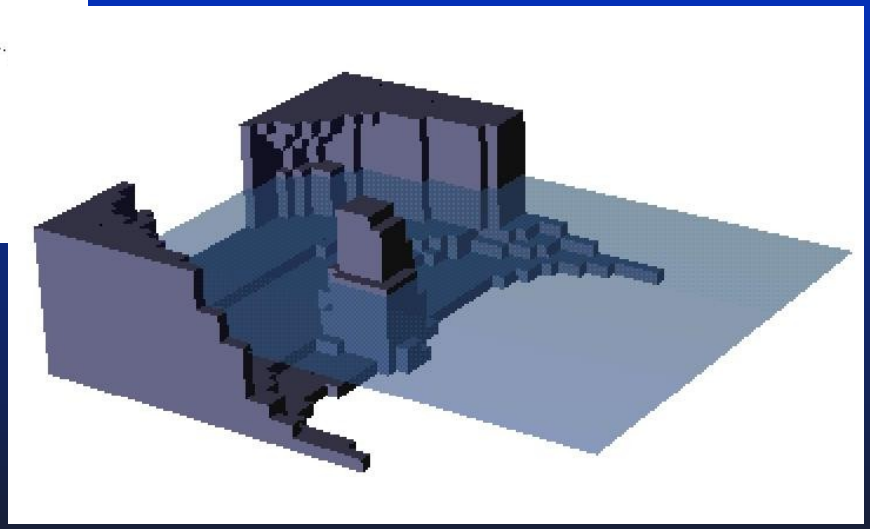




- Couples velocity and pressure in a liquid
- Distinctive motion from incompressibility
- Low energy splashing gives distinctive look
- Easy to solve inefficiently
- Huge literature base (CFD)

Environment Discretization





Timestep Limitation

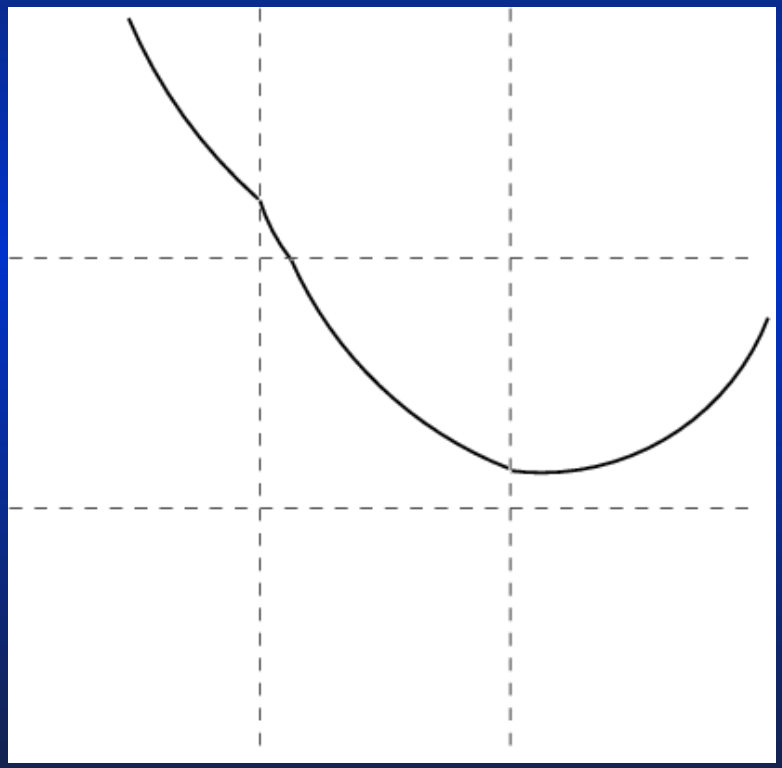
Courant-Friedrichs-Levy (CFL) Condition

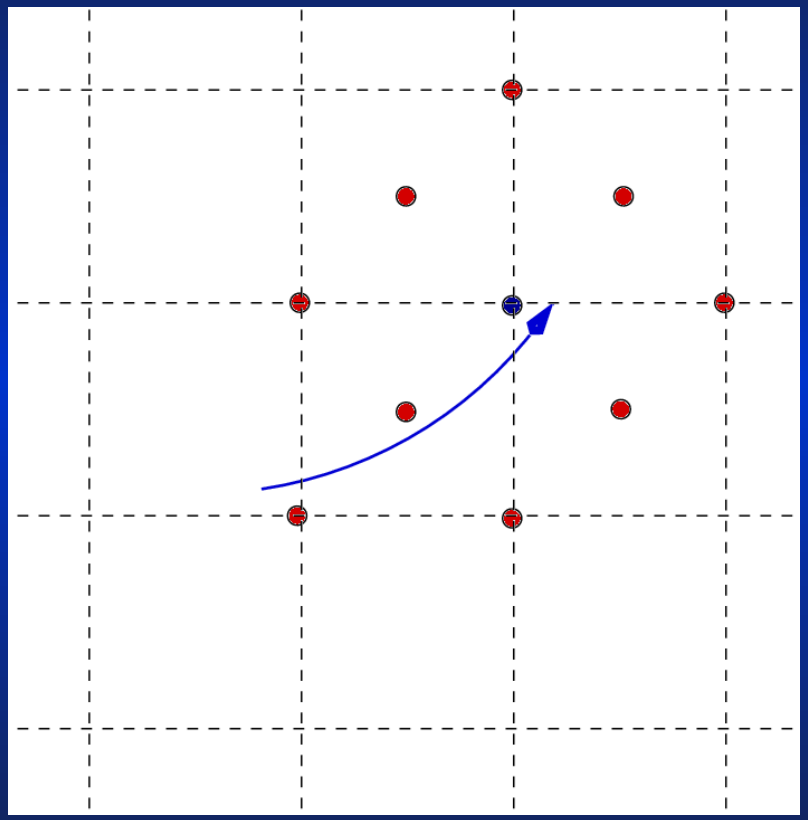
$$\Delta\tau > \Delta t|\mathbf{u}|$$

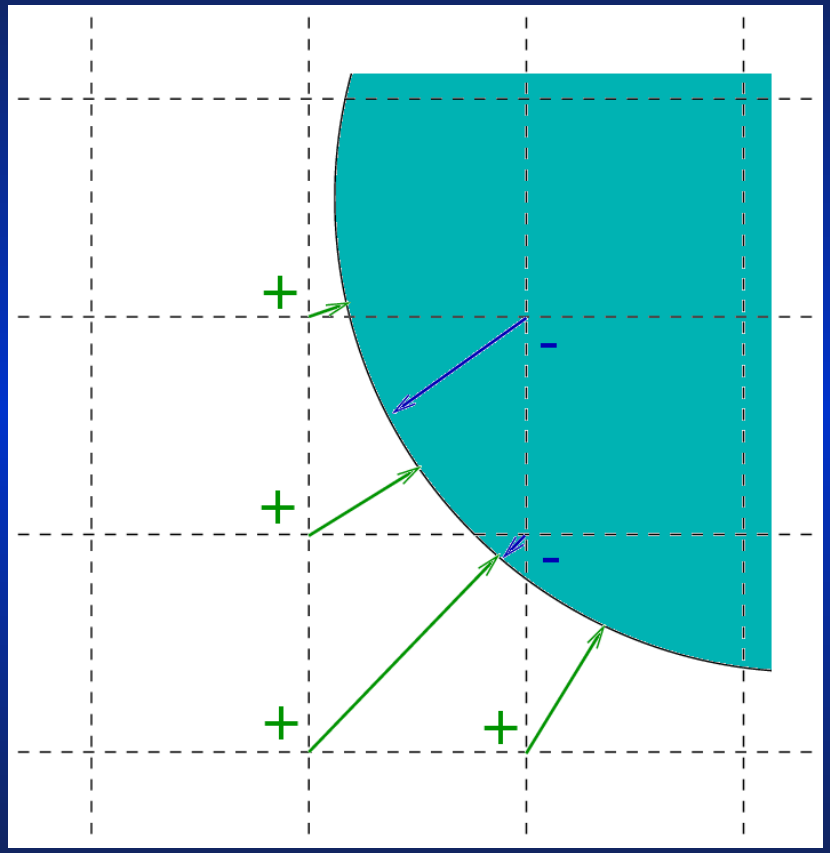
Semi-Lagrangian methods allow for large timesteps but at the cost of reduced rotational motion and increased dissipation

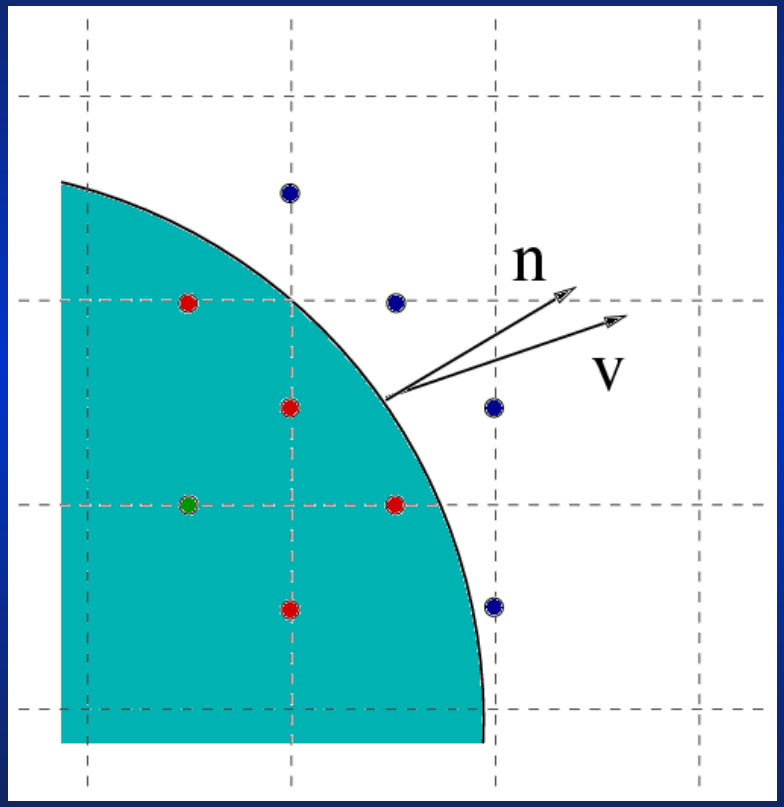
Isocontour

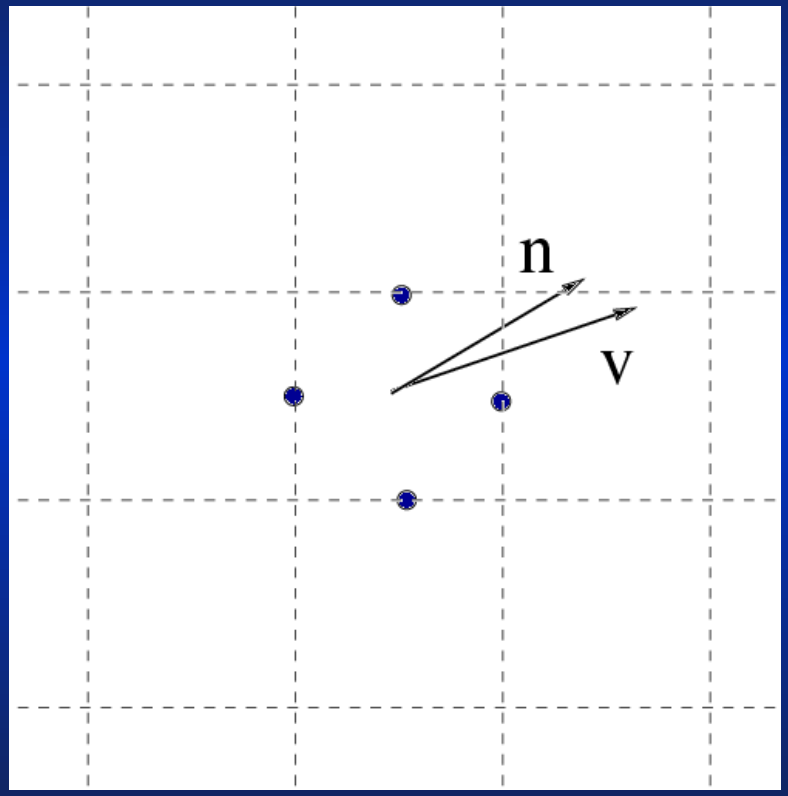
- Eulerian approach
- Easily smoothed
- Resolution limited to grid
- Loses detail during splashing








n
v


n
v

Numerical Stability

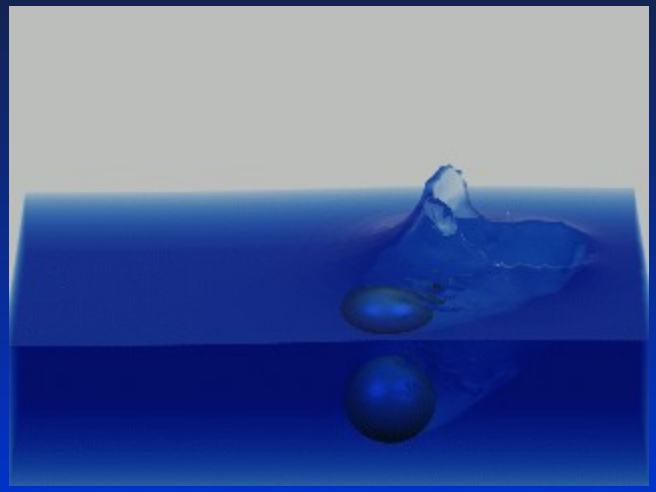
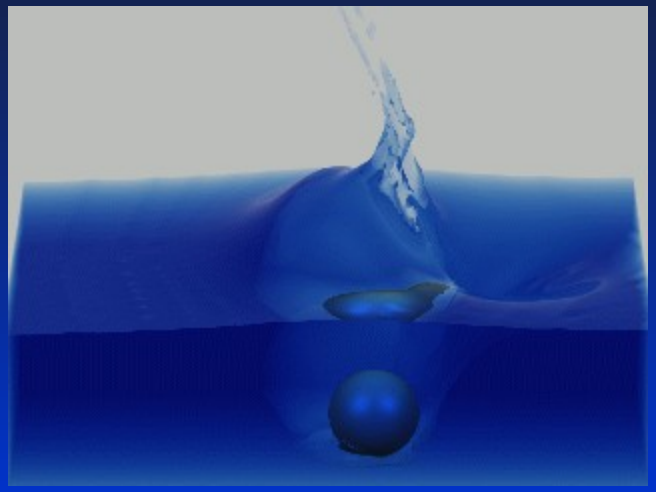
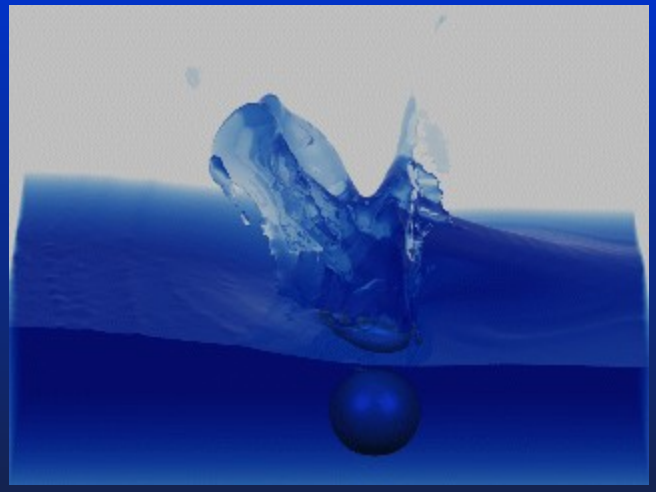
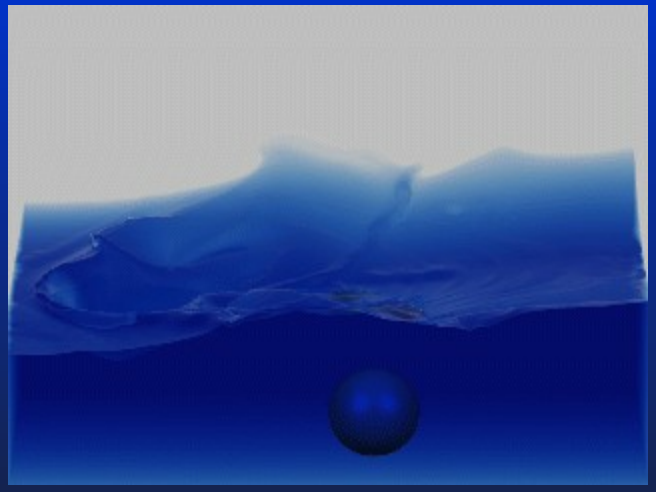
Restrictions on quantities within the system governed
properties of the numerical methods used

$$\nu > \left(\frac{\Delta t}{2}\right) max\left(u_i^2, u_j^2, u_k^2\right) \qquad \text{[central differences]}$$

Restrictions on visual coherency of simulated result

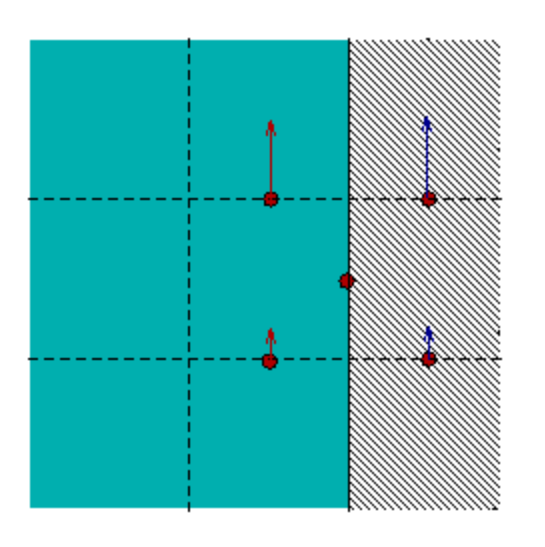
$$\Delta\tau > \Delta t|\mathbf{u}| \qquad \text{[CFL condition]}$$

Critical to understand limitations of techniques used

Object Boundary Conditions

Trick the solver into thinking everything is fl

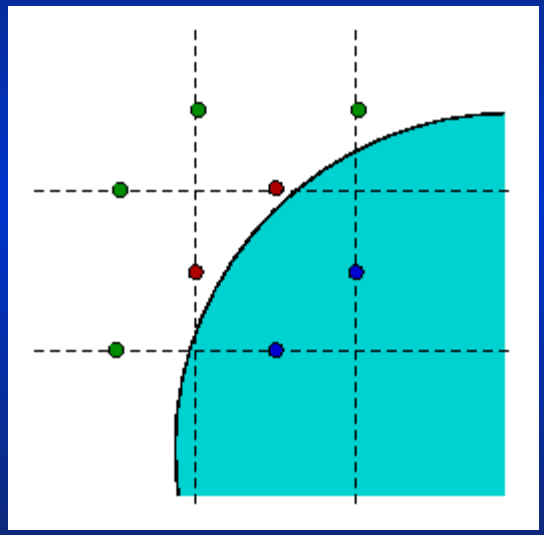


Free slip condition

- Directly set and hold internal velocities
- Pressure need not be set
- Full description in Foster and Metaxas

Free Surface Boundaries

- Satisfy $\nabla \cdot \mathbf{u} = 0$ in the surface cell directly (set red values)
- Copy values out to empty cells (green points) so that particles interpolate correctly
- Set pressure in cell to atmospheric constant



This allows liquid to flow freely into empty cells witho
being “drawn” into or otherwise effected by them

Solving for Pressure

Conserve mass and update the pressure field

$$\Delta p = \frac{\rho}{\Delta t} \nabla \cdot \mathbf{u}$$

Applied again to a cell in the grid

$$\sum_{n=\{ijk\}} \left(p_{n+1} + p_{n-1} \right) - 6p = \rho \frac{\Delta \tau}{\Delta t} \sum_{n=\{ijk\}} \left(u_{n+1} - u_n \right)$$

Adjusting Final Velocities

Once the updated pressure field has been cal
we can adjust velocities to conserve mass

$$u^{t+\Delta t}_{\{ijk\}} = u_{\{ijk\}} - \frac{\Delta t}{\rho \Delta \tau}\left(p_n - p_{n-1} \right)$$

The velocity and pressure fields are now com
for time $t+\Delta t$